The Institution of Engineering and Technology

Intelligent Wireless Communications

Edited by
George Mastorakis, Constandinos X. Mavromoustakis, Jordi Mongay Batalla and Evangelos Pallis

Chapter 5

Intelligent agents system for medical information communication

Mariya Evtimova-Gardair[1] and Evangelos Pallis[2]

This chapter describes the importance of the agents when making an intelligent web communication system for medical information. The state-of-the-art of the agents is performed and the implementation of mobile agents is highlighted with their positive and negative features when gathering information in the Internet. In this respect, an intelligent system with a mobile agent model for communication of medical information is proposed that could be implemented in the wireless structure. The platform for the creation of the medical communication information system that is proposed is a part of intelligent wireless communication. The result from using mobile agent is generated which is also important when working with big data.

5.1 Introduction

In the recent years, there has been growing interest in studying multiagent systems for searching of information. Modern scientific researches which are related with multiagent systems are directed mainly into coordinating agent behavior and distribution and unification of decisions that are applied for information retrieval systems, especially when searching information in a big volume of data. Nowadays, multiagent systems are one developed direction that is applied to systems for searching while preparing theoretical and laboratory research. Agents may have a number of features, such as autonomy, reactivity, proactivity, and social capabilities. These characteristics determine their behavior and interaction with the surrounding environment, which is important for their application in information retrieval systems. The environment is everything external to the agent and the agent is seen in continuous connectivity and interaction with it. The multiagent system is determined by the number of agents interacting with one another. It can be considered that agents interact on behalf of their users with different purposes and

[1]Center for Teaching and Research in Computer Science (CERI), Avignon University, Avignon, France
[2]Department of Electrical and Computer Engineering, Hellenic Mediterranean University, Heraklion, Crete, Greece

motivations. In order to achieve effective interaction with each other, agents must be able to cooperate, coordinate, and negotiate with the people. Multiagent system and its feature determine their application for an information search system:

- Agents have incomplete and contradictory information about the environment and are limited in their capabilities
- Distributed multiagent system management
- Decentralized data
- Asynchronous calculations

Recent research on multiagent systems is mainly related with coordination of the agent behavior and distributing and unifying decisions that are critical when working with big data. Multiagent systems are used in many modern information retrieval systems. The problem of the need to continuously improve the quality of search systems implies the standard for all new information systems to evaluate the quality of returns and compare the result with the quality of returns of other created search engines.

5.2 Analysis of the agent technologies

Searching systems that are on the market have a number of disadvantages and do not always meet the need of the medical information of the user who have a health problem. A solution for an effective significant search system from medical information and Internet is a searching system that uses artificial intelligence.

Multiagent systems and autonomous agents provide a new method for analyzing, designing, and implementing complex applications because they are a part of distributed artificial intelligence (DAI). Today, the majority of applications require distributed tasks between autonomous agents to achieve their goals in an optimal way [1]. This enables intelligent agents to give users accurate and relevant (up-to-date) results. This is the result of the ability of intelligent agents to help the user find and filter the information on the web. This necessitates the use of intelligent agents in search systems, which allows information systems to work with a great deal of information and to enable search systems to learn from their environment in order to reproduce accurate and relevant results [2].

5.2.1 Meaning of the agent when searching information

Nowadays, thousands and millions of people are engaged in searching for information each day when they use a web search engine or search in their Internet mail. Currently, the global search information space consists of millions of HTML pages on the Internet. A big volume of variable, semi- or unstructured heterogeneous data are additionally available in related databases, file systems, multimedia database systems, and software applications. This includes, for example, bibliographic inputs, photos, speaking text, and video data [3–5].

The impacts of data, system, and data heterogeneity of information overloading to the user are numerous. This is particularly true due to potentially

significant differences in the data model, data structures, data representation, using ontologies and dictionaries, query languages, and operations to search, retrieve, and analyze information in an appropriate context.

Information agent technologies are emerging as a major part of intelligent software agent technologies. The leading idea for information agent technologies is the development and efficiency, efficient use of standalone computing software objects called intelligent information agents who have access to multiple, heterogeneous, and geographically distributed information sources, both on the Internet and in a local intranet. The primary task of such agents is to proactively search, maintain, and mediate relevant information with users and other agents. This includes skills such as searching, analyzing, manipulating, and distributing heterogeneous information, as well as visualizing and directing the user to the possible individual information space [6].

5.2.2 Intelligent agents for information: definitions and basic features

Information agents can be classified by type of information software agents. Software agent technologies come from distributed systems with artificial intelligence. The term agent is widespread in the literature. It can be seen as a tool for analyzing systems, not a complete feature that separates the world from agents and nonagents. Intelligent agents are usually assumed to exhibit autonomous behavior that is determined by their:

– proactivity, which means taking an initiative to meet the goals of a design and displaying purposeful behavior;
– reactive and advisory actions mean environmental perception and timely change of management to match design goals; and
– social interaction in groups with other agents and human users if necessary.

This depends on the specific field of application and the potential for solution of a problem, with what type of information agent should be used in practice. Agents are deployed in various environments such as industrial control, Internet search engine, personal assistant, network management, games, distributed software, and more [1,7–11]. Multiagent technologies have standards for software architectures and applications such as OMG MAF and FIPA specifications. Intelligent Internet agents are commonly referred to as an information agent. Information agent can be defined as autonomous, existence of computing software (intelligent agent) that has access to one or more heterogeneous and geographically distributed information resources that proactively use, mediates, and maintains relevant information on behalf of users or preferably exactly during working hours. Information agents are supposed to satisfy one or more of the following requirements:

– Information acquisition and management: There is an opportunity to provide clear access to one or many different information resources for information

retrieval, analysis and filtering of data, monitoring of sources, and updating of relative information as part of its user or other agents.

The acquisition of information encompasses a wide range of scenarios including the advancement of information search in databases and also the purchase of relative information from the supplier from the electronic shopping places.

– Information synthesis and presentation: Agents are able to merge heterogeneous data and provide unified, multidimensional views of relative information to the user.
– The agent can dynamically adapt to changes in user preferences as well as in the network environment and provide intelligent and interactive help to the typical users who support their information business on the Internet. In this context, using the intelligent user interface as a believable, realistic hero can significantly increase not only the user's awareness of his personal information agent but also the way information is interactively previewed. Many systems with information agents have been created or are currently being developed in academic and commercial research laboratories, but they still have to wait to create a realistic world of broad Internet users. The ambitious and required goals to meet all the requirements listed above are supposed to be completed over the next 10 years.

5.2.3 Classification of information agents

Information agents can be characterized in several different classes according to one or more of the following:

1. Noncooperative or cooperative information agents, depending on the ability of agents, cooperate with each other for the performance of their tasks. Several protocols and methods are available to achieve cooperative autonomous information agents and various scenarios, such as hierarchical delegation of tasks, contracts, and decentralized contracting.
2. Adaptive information agents have the ability to abdicate only the changes in the network and the information environment. An example of such a type of agent is the training of personal assistant on the Internet.
3. Rational information agents are utilitarian in the economic sense. They play and can even cooperate to increase their own benefits. One main application domain for such agents is automated trading and e-commerce on the Internet.
4. Mobile information agents have the ability to travel automatically over the Internet. Such agents can allow, for example, dynamic load balancing in large-scale networks, a reduction in data transfer between information server applications, and small business logic migration with middle-range enterprise search intranets [8,12–16].

Advantages of mobile agents [16]:

1. Reduce network load
2. Overcoming network latency
3. Encapsulation of protocols
4. Asynchronous and autonomous implementation

5. Dynamic adaptation
6. Natural heterogeneity and strength
7. Resistance to failure

According to the definition and classification of information agents, they can be divided into communication agents, knowledge, cooperation, and low-level tasks. Figure 5.1 lists the basic skills of the information agent. The communication agent's communication skills define a communication with information systems and databases, human users, or other agents. For the last case, the use of language for the communication agent should be considered at the top of, for example, platforms or specific applications. The information agent acquires and maintains the knowledge of himself and his surroundings in presentation and processes of ontological knowledge and metadata, profiles and input languages, translation of the data format, and application of machine learning techniques.

A high level of cooperation of the information agent with other may lead, for example, to mediation, coordination, negotiation, and cooperation (social) filtering. The intelligent agent, working with its human users, applies techniques that came from human–computer interaction and effective computing.

5.2.4 Basic features of the intelligent agents for information

According to the main properties of the intelligent agents for information, basic supporting technology are defined in the following cases whether they cooperate with other agents or not.

– Access to heterogeneous distributed information systems and resources on the Internet. This includes standardized platforms as well as efficient client techniques of server-side web-based applications.
– Searching for and filtering relative information from any type of digital international environment, such as content-based, rich media, and complex language for searching of information,
– Management of metadata and the process of the ontological processing of knowledge that facilitate the reconciliation of the semantic heterogeneity of retrieved data and information derived from multiple heterogeneous sources.
– Visualization of the information.

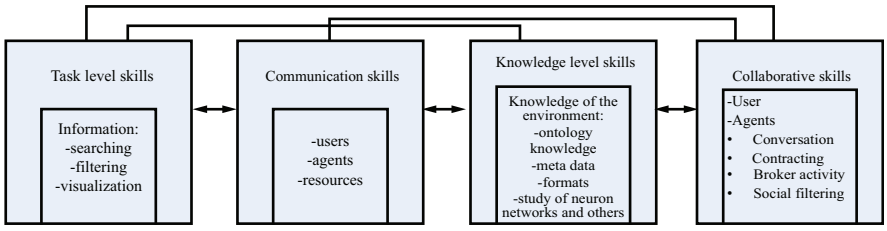

Figure 5.1 Basic skills of the intelligent agent for information

5.3 Usage of multiagent system for extraction of information

5.3.1 The possibilities for the creation of mobile agents

The growing number of applications that serve millions of users and use terabyte data require a paradigm for faster processing. Nowadays, there is a growing interest in large data analysis. Big data analysis is becoming a very important aspect of productivity growth, reliability, and quality of service (QoS). For processing big data, using a powerful machine is not an effective solution.

In the beginning, the semantic web was designed to share knowledge from distributed, dynamic, and heterogeneous sources, in which content is expressed in machine-readable formats through languages such as RDF and OWL [17]. This is the way information is shared on the web. By using agent technology, agents play an integral role using machine-readable formats for gathering and presenting knowledge, as well as reasoning to manage and reach new facts. Together with their ability to process semantic web content, agents contribute features such as distribution, autonomy, and distributed knowledge bases. There are many tools that are designed to manage standard sources of knowledge, but they are usually centralized and in a static environment where overall control is centralized. But such achievements cannot successfully integrate into such an environment as the semantic Internet, which is an open, dynamic, and often chaotic environment [18].

Distributed decentralized systems are characterized by components with the same roles and the ability to exchange knowledge and services directly with each other [19]. P2P systems are equal networks with equal roles and capabilities, and such a system is proposed that uses P2P technology to share and search for a large amount of data [20,21].

5.3.2 Mobile agents

When a distributed web crawler is based on the migration of a web agent or migrating agent [2,22–24], the process of selecting or filtering of the web documents can be performed on the web servers, and not from the side of web searching agents that can reduce network load derived from web robots [23,24]. The mobile agent [25] is a type of software agent with features such as autonomy, social abilities, learning, and mobility. It is a composition of computer software and data that have the ability to migrate from one computer to another autonomously and continue their processes on a remote computer. In the migrating web robots, the mobile code generated from the side of the searching system is executed on the web servers, the environment that is controlled from the other side.

Mobility allows the agent to move or jump between agent platforms. Migration agents include computational software processes that can walk through the most big part of the network, as well as the Internet, interacting with foreign hosts, gathering information on behalf of their owner, and returning once they perform the duties defined by their owner. The agent is an autonomous entity that acts on behalf of others autonomously. Nwana [25] identifies seven types of agents, that is,

collaborative agents, interface agents, reactive agents, hybrid agents, and intelligent agents. Basic features of mobile agents are [2,25] migration, data collection, route determination, and communication.

The agent platform provides a computing environment in which the agent operates. The platform from which the agent originates is defined as the starting platform and is usually the most secure environment for the agent. One or two hosts may include an agent platform, and the agent platform can support multiple computing environments or meeting locations where the agents can interact. They can collaborate or communicate with other agents by locating some internal objects and methods known to other agents without sharing all of the information.

Mobile agents ask their environment to get the information they need to reach their goals. This information needs to be filtered locally by the agent in advance or stored by the agent or forwarded to a receiving destination. When the agent once ends up with the network node, it must decide where to move afterward.

The ability of agents to communicate is fundamental for mobile systems. Advantages of mobile agents are [2,24,25] efficiency, traffic, latency, performance of asynchronous tasks, strength, tolerance to failures, support form for heterogeneous environments, support for e-commerce, paradigm for easy development, and equal (P2P) [20,21,26,27].

When big data are stored on remote hosts, these data must be processed locally instead of being transferred to the network. The basic concept is that it is better to move the calculations to the data than the data to the calculations.

When migrating to the resource location, the mobile agent can interact faster with the resources than over the network.

While the agent acts on behalf of the client on a remote site, the client may perform other tasks. Instead of being online for a long term, the mobile user can create an agent request while it is off, then the agent can be launched during a short session and the agent can accept the results later.

5.3.3 Comparison of the standard model of searching system and searching systems with mobile agents

Web searching system in Figure 5.2 represents a set of programs that can read every page that can be found on the Internet, creates an index of the information that found, compares the information with that of the user's query, and finally returns the results again to the user. It is a searchable database that collects information from web pages, indexes the information, and then saves the result in a big database where it can be searched quickly. Web search engines are a link between the web user and web documents. Without the help of search engines, endless information on web pages remains inaccessible to the user. Web search system basically consists of three parts: web bot, index, and query mechanism.

The web robot represents a module that searches web pages from the web. These are small programs that read on the web on behalf of the web search engine and follow the links to reach the different pages. As it is started from selected URLs, robots retrieve URLs that appear in retrieved pages and keep pages in a

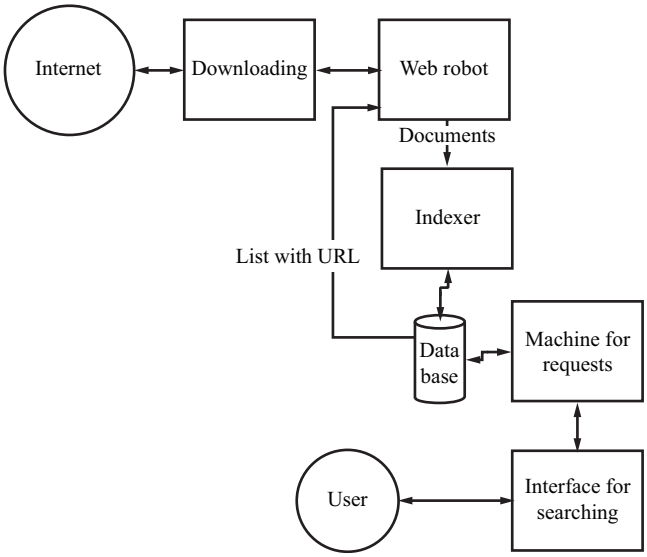

Figure 5.2 Principal architecture of the web search

warehouse database. The index extracts all unusual words from each page and registers the URL to which each new word is found. The index retrieves all unusual words from each page and registers the URL where each word is found. The result is stored in a large list containing URLs pointing to pages in a database where a word is encountered. The machine for querying is responsible for receiving and completing user search queries. It relies on the indexes and the database. Because of the size of the Internet, and the fact that users usually put one or two keywords, the result of the information found is usually quite big. Methods with distributed web robots with migrating agents [2,22,23,28] allow packet wrapping and distribution to specific hosts where interaction can take on the same place. Migrating agents are also useful in reducing the flow of raw data on the network. When a big amount of data is stored on remote hosts, these data must be handled locally where data are located and not transmitted over the network. The basic concept is the movement of calculations to data rather than data to calculations. When migrating to the resource location, the mobile agent can interact with the resource much faster than over the network, and also reduces network traffic.

A distributed web-based method with migrating agents is presented in Figure 5.3, which uses web-based website management, which assigns migrating web robots to web servers a list of URLs on the respective web servers. Migrating robots, when they reach the server, crawl the pages, choose the best of their collection pages, and return to the web search engine with the collection. This reduces the unnecessary amount of information and therefore unnecessary pages of the web search engine. The size of the collection can be further reduced by filtering the required specialized web pages and even compressing them.

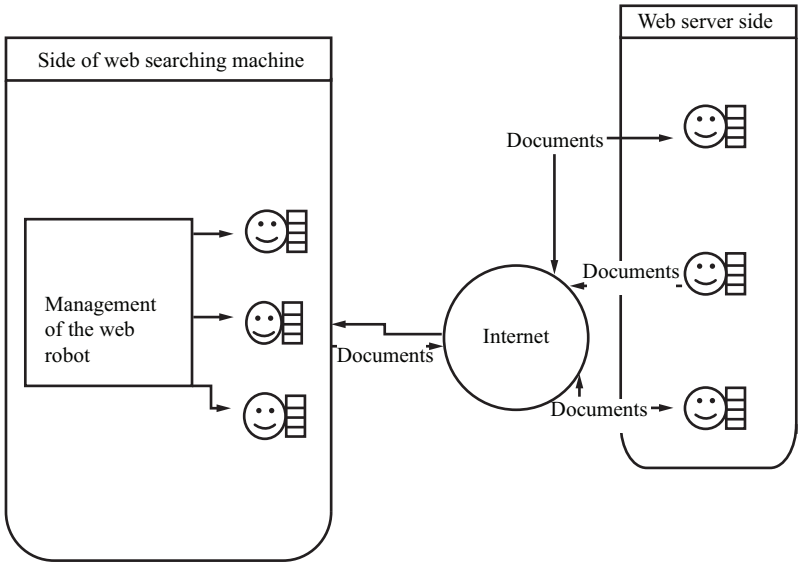

Figure 5.3 Web searching system based on agent migration

5.3.4 Features and benefits of mobile agents

A mobile agent is a software abstraction that can migrate during execution through a heterogeneous or homogeneous network. It has the ability to stop its performance according to some factors and to resume it in another machine.

Mobile agent features: There are several features that can determine the structure of the mobile agent.

Condition: A basic feature of the mobile agent. It can stop the execution of a machine and resume the execution of another machine.

The condition depends on two factors:

1. Execution state, which is a state during work that includes the counter in the program and the stack.
2. The status of the object that stores the current values of its variable.

A mobile agent is a software abstraction that can migrate during execution through a heterogeneous or homogeneous network. It has the ability to stop its performance according to some factors and to resume it in another machine. Mobile agent features: There are several features that can determine the structure of the mobile agent:

Condition: A basic feature of the mobile agent. It can stop the execution of a machine and resume the execution of another machine.

Condition depends on two factors:

1. Performance state, which is a state during work that includes the program counter and the stack.

2. The status of the object that stores the current values of its variable.

Execution: This is a program code that defines the behavior of the tasks. If JADE is used as a platform for mobile agents, the classes represent the execution code. There are two ways to make the classes available to the mobile agent [26]:

1. Getting all the required classes along its route and using it any-time anywhere.
2. Taking a part of the required classes and when the mobile agent needs a class that is not available, he can retrieve it from the remote location. This operation is called code-on-demand technique and is a common technique for distributed network systems.

Interface: The mobile agent cooperates with other agents to handle the assigned job. Interface is needed to make communication possible between agents.

Unique identifier: It identifies the agent during its life. It is used as a key that needs to be especially relevant to a specific agent when it is traveling on the net.

Route: This is a group of addresses created once when the mobile agent starts, which defines the agent's route on the network.

Principles: This is the information about the person, organization, or corporation to which the mobile agent belongs. The principles are needed to authenticate the mobile agent who is traveling to several destinations on the network.

Advantages of mobile agents [29]: There are many advantages when using mobile agents for solving problems in distributed application.

Reducing network traffic: Collaboration in a distributed system is often achieved through the use of communication protocols. These protocols transfer a large amount of data stored in remote hosts through the network to a central processing location resulting from high network traffic. In this case, the mobile agent uses alternative communication protocols.

Offline tasks: The network connection may fail at any time.

Agents can solve this problem by performing offline tasks and sending results to a server application when it is online again.

Support for heterogeneous environments: Mobile agents can work on the top of any operating system that has the same mobile framing.

Failure: Mobile agents react dynamically and autonomously to changes in their environment. If a host is stopped or a platform does not work, all agents running on this machine will be alerted and given time to send themselves and continue their operation to another host on the network.

Encapsulation protocol: The encapsulation protocol allows the components of the distributed system to communicate and coordinate their actions. Mobile agents provide a solution to the problem of improving the protocol code at all locations in the distributed system.

The result of using mobile agents in a personalized search engine is to improve the speed of the performance of the system.

5.4 Big data impact when searching

The data have been rising rapidly over the last decades, and "big data" is becoming a popular topic. Big data not only refers to massive data but also to a series of techniques that turn the data stream into valuable information [30]. Big data is related to a large volume, complex, and growing data set with multiple, autonomous sources [31]. "Big data" is characterized by volume, variety, and speed. Big data is not only large in volume but also varied in type. They can be structured data, unstructured data, or semistructured data. Speed describes data flow that requires timeliness in data analysis. The ultimate goal of data analysis is to obtain valuable information from the data stream.

5.4.1 Challenges when analyzing big data

1. Content and search when making analysis

   Traditional data analysis methods developed for structured databases are path analysis, time analysis, charting and network analysis, and what if analysis [30]. Big data are developed rapidly in every aspect of data processing, that is, data collection, data mining, data storage, data modeling, data processing, and data interpretation. With the development of theoretical analyses and technologies, big data will be used in more and more areas (i.e., health, smart grids, the economy, and social networks). In addition to the challenges outlined above, there are still other issues [30]. The security of big data is a difficult problem that needs to be defined by the law and protected by technical measures [32].
2. Diversity and heterogeneity of data

   There are traditional tools, such as SQL, to handle structured databases, but when databases are semistructured or unstructured, such as audio, video, text, and web pages, traditional tools are not effective. Therefore, initially, the data should be structured so that data can be processed more efficiently. To customize medicine, clinical information (such as medical diagnosis, medical images, and patient history) and biological data (i.e., gene, protein sequences, functions, biological process, and pathways) have to be managed and integrated with a variety of formats and are generated from different heterogeneous sources. Over the last decade, it is challenging to work with big data, but today it is important to focus on developing tools and techniques to get a better sense of data and make use of knowledge to find knowledge. Despite the overall availability of data arrays, interoperability is still lacking. In order to personalize data in the basic is the integration of data and the use of different data sources.
3. Volume and size of data

   Big data expands its volume faster than the computational resources. Processors follow the Moore's law until the size of the data leads to an explosion. But there is a limit to Moore's law. Because the size of the chips is becoming smaller and smaller, the quantum effects are so significant that they

cannot be ignored. As a result, a solution needs to be found to overcome the problem with the fast-growing data. The huge volume of data itself does not solve the problems with the quality of the returned results from the searching system. Data must be summarized or retrieved in a meaningful way in order to transform the data into information, knowledge, and finally into wisdom. The big volume of data needs to be effectively explored so that to be used to make a decision. For the processing of big data, tools such as Hadoop is used that can increase the velocity of processing data and queries.

4. Velocity and timeliness of the data

It is accepted that as much bigger the volume of the data is, as much longer it takes to analyze the data. Over time, the data value is destroyed and in some cases, the timeliness of the data processing is important (i.e., bank credit verification). In other words, velocity is a constant that constantly changes and develops when performing healthcare data. These rapid changes in data give a significant challenge in creating the relevant models in the on-demand domain so that to be useful for searching, browsing, and analyzing content in real-time. This requires solving the following questions:

1. Ability to filter, prioritize, and classify data (which refers to a domain or a case)
2. Ability to receive data quickly
3. Ability to select, create, and refine relevant knowledge [33]

Free space, uncertainty, and incomplete data are critical for applications with big data. For data that are rare, the number of data points is too small to produce reliable conclusions. This is usually a compilation of data size problems where data in high-dimensional space (such as more than 1000 dimensions) do not clearly show directions or distributions. For most machine learning algorithms and algorithms for data mining, high-dimensional free data significantly degrades the reliability of the models derived from the data. Such studies are created to use the reduction of the dimensions or the selection of properties to reduce the size of data or to carefully additional examples to reduce the lack of data such as generic uncontrolled training methods when searching in data. Uncertain data are a special type or real data, where each data field is no longer determined but is subject to sporadically appearance of error. This is mainly related to domain-specific applications with inaccurate data and collections. The use of complex data is a major challenge in big data applications because every two sides in a complex network are potentially mutually interested in making a social connection. Data from application that engages in dynamic environments have many sources, massive, heterogeneous, and dynamic characteristics. This determines the important features of big data, such as petabyte (PB) calculations and even the exa-byte (EB), a level of complex calculation process [31,34].

The concept of big data quickly expands in many sciences and engineering fields. Free space, uncertainty, and incomplete data are critical elements for big data usage. Consequently, the use of parallel computing infrastructure and,

respectively, programming language support and software models for efficient analysis and searching of distributed data are essential objectives for big data processes to switch from "quantity" to "quality" [14,35].

5.5 Model of the multiagent searching system

5.5.1 Requirements of the searched system

The proposed searched system is designed to provide a quality search for medical information that corresponds to the individual needs and needs of people of all ages who would like to receive information on a health problem depending on their medical complaints. Because English is the most widely used language and could be found the most and best quality text sources in any field, the system will be developed and tested in English for the time being. The approaches, methods, and algorithms used in it could, in principle, be implemented in any natural language, but this will require serious efforts because of the significant differences between human languages and, moreover, the effectiveness of the system depends on the text and sources, meaning that it would be most effective in English at the moment.

Main system requirements are as follows:

- Ability to display user information on given search words
- Adaptability—easily adapted to the needs of the particular user
- Expandability—it is easy to add new components to the system
- Work effectively with vague and inaccurate information from the user
- Ability to work with big data
- Precision of Internet search according to the specific case
- Increase of the recall results from the search according to the specific case
- User-friendly interface
- Platform independence requirements of the searched system

5.5.2 The choice of multiagent platform for system realization

An agent could be defined as an entity that performs one or several tasks to be able to distinguish their goals. Some of the special characteristics of the mobile agent are move between several machines changing its state, possibility to find by themselves the way on the network, and migration from one execution environment to another. Personalization of the searching results could be performed with semantic rules. The choice of a multiagent software platform is important for the creation of multiagent software. Some of the agent platforms are no longer maintained and others continue to be used.

There is a variety of software for multiagent systems that are either commercial or open-source [29]. Also, there is, in Wikipedia, comparative analysis of some basic characteristics of multiagent systems [36].

One of the basic characteristics is reviewed in the literature sources for comparison of multiagent software [29,37–39]. Concerning the development of a multiagent system appropriate for the field is reviewed in the following criteria:

- Programming language for the creation of the specific MAS
- Standards requirements supported by the MAS
- Tools for communication
- Mobility of the agents—strong (possibility of the system to migrate code or condition of execution from the code); weak (migration only of the code)—clean and effective method for migration, threads must be recreated or restarted from waiting daemon.
- Transport of the messages
- Security
- Accessibility
- Model
- Elements
- Level of activity
- GUI tools
- Specifics

5.5.2.1 Aglets multiagent platform

Aglets (http://aglets.sourceforge.net/) was created by IBM in 1997 and is open-source since 2001, which is a very popular mobile agent platform. But there is no new release until 2004, so the future is not clear. One of the advantages is that it follows MASIF specification. Aglet is created from the one thread model of agents and a communication infrastructure based on the transmission of message [38].

- Java
- Integrated standard MASIF works with CORBA
- Sockets: exchange of messages, maintain proxy, but not dynamic proxy; maintain synchronous and asynchronous communication
- Weak mobility
- Asynchronous transmission of messages; during synchronous transmission of messages deadlock is possible
- Basic security mechanism: security during inter-platform communication
- Open source from IBM
- Events model
- Elements: contexts, Aglets, and Tahiti
- Level of activity—very weak
- GUI tools available
- ATP (maintain HTTP sets, problems with firewall; one-way without sending); initialization

5.5.2.2 Voyager multiagent platform

Voyager (http://www.recursionsw.com/) was created in the beginning from ObjectSpace in 1997 and now from Recursion Software, the computing platform is

distributed, that is directed mainly in simplification of the management of remote communications of the traditional CORBA and RMI protocols [38].

- Java
- CORBA and RMI protocols
- Maintain proxy; doesn't maintain messages
- Strong mobility
- Synchronous and asynchronous communication
- Security mechanism (management of the security and so on)
- Commercial product: paid
- Procedural model
- Elements: servers; agents;
- Level of activity: high;
- No available GUI tools
- Features: multicast; publish/ recording; and dynamic aggregation

5.5.2.3 Grasshopper multiagent platform

Grasshopper is developed from IKV++ in 1999 and then it became a part of the commercial Enago Mobile and now its development is left [10].

- Java
- Maintain the following standards: MASIF, FIPA, and CORBA working (VisiBroker and Orbix)
- Perform weak mobility with the ability to maintain strong mobility
- Communication: ACL, synchrony, asynchrony, dynamic, multicast communication, and different transport protocols (sockets, RMI, and IIOP)
- Basic security: policy for security: external (X.509, SSL-confidentiality, integrity of the data, and general certificate) and internal security (based on the mechanism provided from JDK)
- The product is not available
- Procedural model
- Elements-: places, regions, and agents
- No level of activity
- There are graphical tools
- Features: MASIF, FIPA, and multicast

5.5.2.4 JADE platform

JADE is developed from Telecom Italia Lab in July 1998 and is open source from February 2000. This is a very popular FIPA helpful agent platform. The agent consists of different concurrent behavior that can be added dynamically. One of the benefits is that there is a wide range of tools and can be integrated with other software such as Jess (rules generator). It must be mentioned that it maintains the development for the representation of knowledge of the agents [38]. Furthermore, it is applied to the full FIPA communication model. There are software instruments for the correction of mistakes during development. Configuration could be changed during implementation by moving agents from one computer to another [1].

Mobility is not a key element from JADE and maintains mobility between containers.

JADE maintains mobility between containers in the same JADE platform (like the idea for containers in Grasshopper) [38].

- Java
- Integrated standards: FIPA, works with CORBA (Orbacus)
- Communication: ACL, maintains for interplatform messages with the plug-ins MTPs (RMI, IIOP, HTTP, and WAP), ACL XML codec for messages
- Mobility: weak mobility
- Policy for security: Object manager in Jade provides validation of the connection, customer validation, and RPC encryption of the messages. JADE socket proxy agent holds like bi-directional way between the JADE platform and the TCP/IP connection
- Open source platform
- Model: behavior
- Elements: containers, basic containers, platforms, agents, DF, AMS, and MTS
- Level of activity: high
- Available GUI tools
- Features: FIPA, JESS, JADEX, maintains ontologies

Quality estimation compared to other platforms shows the advantages of the JADE Aj(Amax) = 33, Aj/Amax. 100, and % = 84.6%. Comparative analysis of the existing mobile agent platforms that are most famous define that SPRING have the best parameters [40], but one of the basic disadvantages is that it does not support FIPA standard. Concerning to the report [40], it is defined as more appropriate for the choice of mobile platform Glasshoper, JADE, and AGLETS, as it follows the line of order. But for Glasshoper, it must be added FIPA, so that it is able to use it and AGLETS does not maintain FIPA standard. JADE platform is very well documented and has strong support from the industry and also a broad user community [39]. Practical usage of JADE: a lot of universities and companies including INRIA, Nice-Sophia-Antipolis, ACACIA research team, ATOS Sophia Antipolis agency in the European COMMA project, KOD project IST-12503, CSELT, KPN and Starlab in DICEMAN project, and business and technology research laboratories.

JADE

JADE, the multiagent system, provides the application that will develop the possibility to work in nonpredictably changed web environment. JADE provides efficient and standardized way for mutual cooperation and coordinates the work of the agents through the exchange of standardized ACL messages. JADE clearly differentiates the process for building from the process for using the profile. JADE gives relative autonomy to the system for building the profile and in the same time takes and controls all nondeterministic and time and resource consuming operations and in this way provides effective work of the search system and the possibility to react and return the found results in acceptable consumer time. Basic disadvantage in

JADE is that it use ontologies only for providing a common dictionary in the process of communication between the agents and cannot automatically develop ontologies. So that to be able to use JADE for dynamic creation of web ontologies, it is important to put extension of the multiagent platform that adds the possibility to work with protégé ontologies [41]. Each container in JADE includes directory to facilitate DF agent, remote monitoring interface (RMI), and agent management system (AMS). In this way, agents can find other agents dynamically using DF agent and can communicate with each other using peer to peer. JADE agents communicate using asynchronous transmission of messages that is mostly widespread for distributed and free connected communications. JADE defends security of the agents who provide a serious certified mechanism to confirm the rights of each agent. Messages that are exchanged between agents use agent communication language (ACL) defined by FIPA. Even so, JADE maintains the implementation of multiple parallel tasks with one thread. This characteristic maintains the resource limit of the environment. This maintains agent mobility that let the agent transfer her code and also the condition of the remote hosts. Mobility in JADE is well known as "not-so-weak" as steak and program counters are not able to be saved in Java. JADE has integrated security features. JADE object manager provides connection, validation of the customer, and RPC encryption of the message. JADE socket proxy agent maintains a bidirectional entrance between the JADE platform and the common TCP/IP connection [38].

JADE maintains a protocol contract net to facilitate a difficult multiagent application. JADE can manipulate internal security with other agents using standard FIPA.

5.6 Conceptual schema of the proposed searching system

The conceptual schema defined in Figure 5.4 can find relevant medical information that corresponds to the customer health problem [42–44]. That conceptual schema is presented in two basic parts. The first part includes the definition of the query that is requested from the user with personalization. And the second part is the mechanism of searching information from Internet sources when using a mobile coordinating agent. Agents represent software elements that have their goals, and can operate autonomously in a particular environment and can have communication with another agent or group of agents. Using agent-based programming has several advantages. The agent can communicate with the environment that leads to update the user preference dynamically. In addition, the system requires a lot of interactions with the user and the system to define the query and return appropriate results. Furthermore, the agents are recommended to be used because of their effective communication and level of abstraction. In the proposed conceptual schema, the user first creates a profile where he can add common information, medical information, and health condition and after that, the user enters text questions into the interface. After that, the user request is modified with the content of the user profile so that the results become personalized concerning the user request.

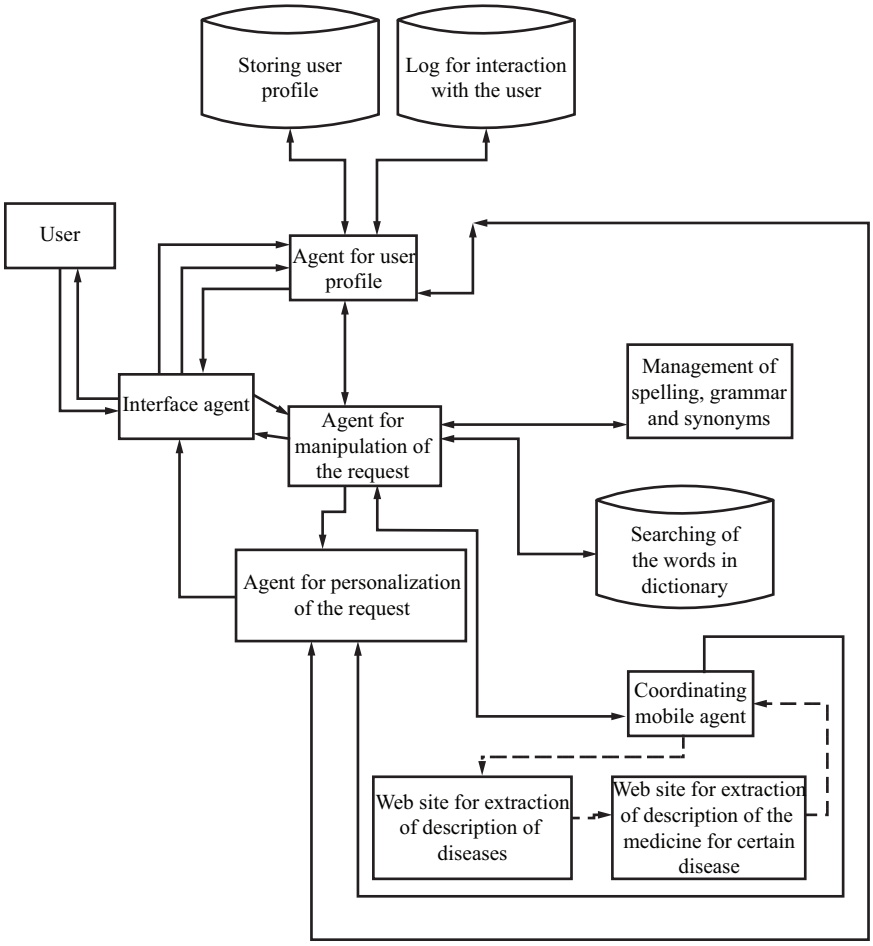

Figure 5.4 Conceptual schema of the proposed searching system with mobile agent

The conceptual scheme consists of five agents. The interface agent that interacts with the user, agent for user profile that captures and maintains user preferences, agent for manipulation of the query, agent for personalization of the returned results, and a mobile agent that searches information on the websites. In Figure 5.4, the conceptual scheme is shown that is proposed in this chapter.

1. Interface agent

   The interface agent makes the interaction with the user to retrieve the user input and display the retrieved results. The entered data can have the preferences explicitly entered from the user. These preferences can be transmitted to the user profile agent to update the user profile. The entered data could be the user requests that are transmitted to the agent that defines the request to

perform the relevant adjustment of the query. Agent for manipulation of the query will communicate back to the user, whether it is necessary to review the user query and get any missing information or spell correction. Moreover, the interface agent shows the user results and formulates the personalized user results. User interactions may be either explicit or implicit. Explicit interactions can be captured by asking the user about his feedback on the results, while implicit interactions can be captured by monitoring user behavior of results.

2. User profile agent

   Agent of the user profile needs to manage the profile. The user is asked to fill out a form that reflects user preferences. Although many users do not like it, they need to take time to fill in such forms. So that it helps to process the results for the personalization of the results from the agent.

3. Agent for manipulation of the results

   Agent manipulates the results which is required to enrich and process the user request. After receiving the user request, the agent determines the language because each language has its own syntax and processing. The word request is then divided and then the spelling of the query is checked using spell check and synonym manager services. Spellcheck and synonyms manager provides two basic services. One of them verifies the spelling and the other provides synonyms for each term. After that, the query is classified using the word search service in the dictionary. The dictionary for searching words is a predefined term storage that helps to identify stop words and the relationship between the terms. This is a multilingual vocabulary where there is a list of terms in each language that is used to search for and match the terms of the query. After that, the unnecessary words are filtered using a word search dictionary that has a list of redundant words such as- between, do, on, etc. Other terms are then identified as possible links between these terms, using a Word Search Dictionary that matches predefined terms. The agent also gets the synonyms of the spell checker terms and the synonym manager and links them to the terms of the user request. After that, the information is submitted to the mobile agent to find the relevant terms in the request on the relevant web site with information.

   The information gathered from the mobile agent sends the data to the agent to personalize the results. If the user request does not match, then the agent reconsiders the request using the user profile. The profile is retrieved by the agent for two reasons:

   1. To reconsider the request
   2. To add more user information to the request

   Finally, the application annotation is produced and sent to the coordinator agent.

4. Agent for personalization of the results

   The agent responsible for personalization of the results is important for the returned results to the user. Firstly, it receives an annotated request from the agent for manipulation of the query and then applies the appropriate algorithm

to extract the appropriate result of the requested query. In this agent, results are further processed by allocating for conflicts, aggregating similar results, ranking, and sorting after receiving the user preferences taken by the profile agent. Finally, the results are personalized and sent to the interface agent to display to the end user.

5. Coordinating mobile agent

   The function of the mobile agent is to get the requested query from the user and o find the relevant information on the web for this request from the user and then to send this information to the agent for the personalization of the request. The goal of the mobile agent is to get the updated information about a user request and to search for relevant information from websites. After the agent for modification of the request receives a message, it searches to send the request to the coordinating mobile agent. The coordinated mobile agent retrieves the desired categories from the content of the request and searches for all the locations in those categories. Once they are discovered, the movement begins from one place to another. Each place has a corresponding web site, so when the agent arrives at the site, it starts collecting the data from the relevant page and gathers them in a database and then moves to the next place. This action is repeated until all places are visited and then the mobile coordinating agent informs the agent for modification of the query that the data has been collected.

5.6.1 Searching with a personalized search system of information on the web when using coordinating mobile agent

Functionality of the agent includes learning, planning, and searching for the current information in the Internet. Collecting information is a difficult process depending on the type of information that needs to be collected, but researchers try to improve current methods or even find new ones. The coordinating mobile agent accepts the user request from the agent, modifies the query, finds the required category, and transfers the request to the appropriate web agent and that should be used if the agent was not mobile. This diagram is shown in Figure 5.5. In Figure 5.6, a schema is shown with coordinating mobile agent that moves through all locations in these categories. When the agent arrives in a definite place, the mobile agent extracts all the requested information from the relevant web page. The scheme with the mobile coordinator agent is shown in Figure 5.6.

Using a coordinating mobile agent saves the creation of web agents when making the system model simpler.

5.6.2 Presentation of the static and mobile agent when searching for information on the Internet

The sniffer agent in JADE allows real-time visualization about the agent interaction to solve a problem. This sequence is described in Figure 5.7 and it presents the search of information on the Internet when using static agent.

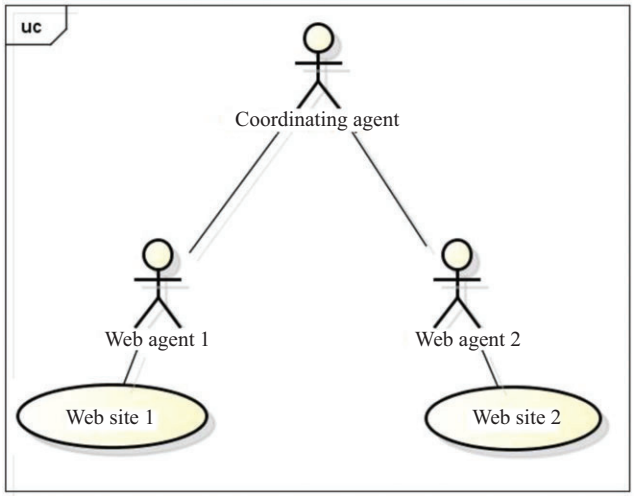

Figure 5.5 Collecting information with static agent

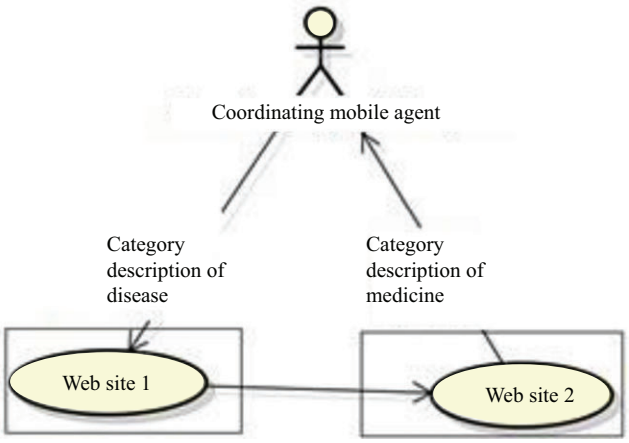

Figure 5.6 Collecting information with mobile agent

When using a mobile agent, its functionality replaces that of the coordinating agent and web agents, making the scheme simpler. As it could be observed in Figure 5.7, when the agent for modification of the request accepts a request from the user. It searches for an agent that offers the coordination service through the request message and receives a response by informing the communication activity.

After that, it sends a request to a coordinating agent who will ask for the agents located in the desired categories. And then it sends a request to a coordinating agent who will ask for the agents located in the desired categories. Once found, the

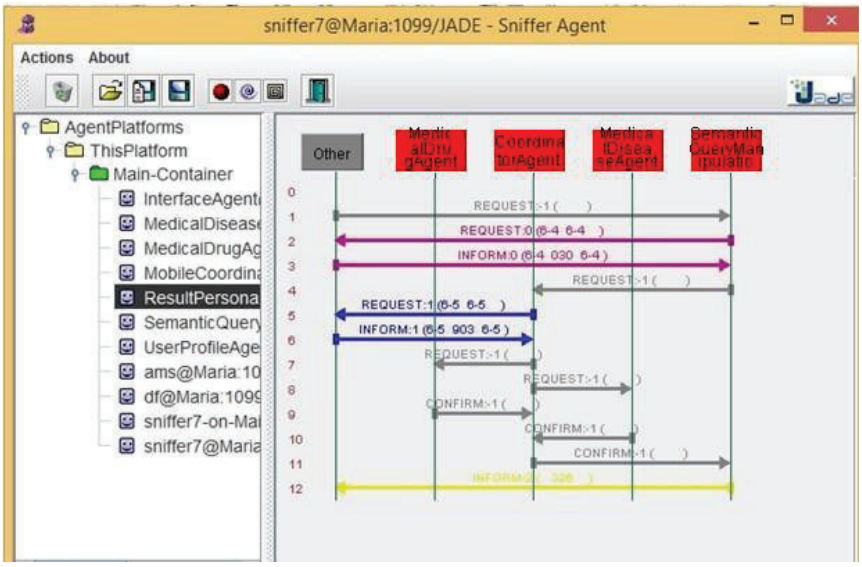

Figure 5.7 The sniffer agent tracks the sequence of the communication between the agents when using static agent

coordinator will send requests to those funds awaiting their confirmation. When all of them have confirmed that they will send a confirmation message to the agent for modification of the request and there is the possibility to learn when the request is completed.

5.7 Implementation of the searching system in Internet

The application is implemented using Java technologies such as JADE and HtmlUnit. The JADE function for agent execution is used. During startup, each agent registers at AMS and then registers his service with DF yellow pages. The services are distinguished by their description so that each web agent is registered with the name of his category, and the agents register with their services. In this way, agents can easily find interaction and communication to achieve their goals.

When registering agent services, "type" is the type of the agent service and "name" is the name of the service. The mobile coordinating agent can register as a "coordinator." The mobile coordinator agent replaces the role of the web agent (using a static coordinating agent) and retrieves information from a specific web page. This process is done using the getResults() method of the class of the corresponding web page.

This method uses HtmlUnit class libraries to fill the search form of the page, click the send button, and then analyze the resulting HTML page to collect the data.

The JADE platform is used to written the following code in the JAVA language. JADE demonstrates the algorithm that searches for the www.medicine.com disease description site.

The collection of data is presented using HtmlUnit with the following code:

```
public void getFinalres(){
WebClient1 webClient1= new WebClient1();
webClient1.setJavaScriptEnabled(false);
HtmlPage1 pages=null;
try{
pages= webClient1.getPage(webSite1);
}catch(Exception el){
el.printStackTrace();
}
HtmlForm1 forms=(HtmlForm1)pages.getElemById("search_fields");
HTMLElement1 destTB=forms.getElementById("DiseaseSearch");
HtmlElement1 checkbox= forms.getElemById("ViewList");
HtmlElement1 submitBut-ton=forms.getElemById("submitButton");
destinationTB.setAttribute("value", disease);
HtmlPage1 finalPage=null;
try{
finalPage=submitButton1.click();
}catch(IOException el)
{
el.printStackTrace();
}
}
```

In the same way, information about the medicine for the relevant disease is also collected from http://www.drugs.com/.

The coordinating mobile agent combines the role of the coordinating static agent and web agent. The coordinating mobile agent first finds all available places in the platform through GetAvailableLocations behavior by submitting a request to the agent management service.

After receiving the request, he will filter the list of places and choose only those who need to use the method for filtering the locations, and then start moving to the websites using the move() method. The method that is responsible after the movement tells the agent what to do after the agent arrives.

Depending on the place where it arrives, it will begin collecting data from the web site. This is repeated until all places are visited and he informs the agent handling the request that he has to complete the task.

5.8 Implementation of the intelligent system in Internet

In order to justify the use of mobile agent when searching for information, a short comparative analysis with results is presented. This is shown in Table 5.1. These

Table 5.1 Comparative results

	Searching with static agent	**Searching with mobile agent**
Time for response	80544 ms	71123 ms

results are obtained by performing the searching of data in the Internet with a mobile or static agent. For system testing, a computer with the following system configuration is used: Intel Core i7-10510U @2.20GHz, 16GB RAM. A detailed description of the results obtained is presented in the following paragraphs.

The total time it takes to get a response from static agent architecture will be the maximum time it takes for a web agent to collect all the data from the web page plus the time for agent communication. Compared to this, the total time required for mobile agent architecture will be the sum of the times needed to collect the data from each web page. The time required for communication is reduced in mobile agent architecture because of the reduced number of agents involved.

At first glance, it seems obvious that the system works better with static agents, but actually when making analysis it observed different results. The larger number of agents running the system at the same time on the same platform, even those waiting, presents a large number of computer processes that need to be handled, which makes it harder for agents to work and filter web pages when looking for the desired information. This aspect makes the period of time used to collect data grows, resulting in a slower system. Based on these results, mobile agent architecture is more preferred because of the need for resource management. Another advantage that can be added is the security of the communication channel communicated by the agents. The test results can be changed when the application of a different system configuration is started at another speed of the Internet connection. So the mobile agent architecture approach has advantages in the current context: the way data are collected in combination with the system, but when information gathering is improved on the side of the information, there is a good chance that a static agent approach is better.

5.9 Conclusion

This chapter describes an intelligent system with a mobile agent for communication of medical information. The system after testing gave qualified returned results to the user compared with the other intelligent systems for information of medicine for a reasonable time when using a mobile agent. The time for searching is very important when working with big data. Nowadays, more and more people, professionals and nonprofessionals, are searching for medical information in the Internet and the proposed intelligent mobile agent system will help them to select the appropriate medical information they need.

References

[1] Mavromoustakis, C.X., Pallis, E., and Mastorakis, G. *Resource management in mobile computing environments*. Vol. 3. Modeling and Optimization in Science and Technologies Book Series. Cham: Springer; 2014. p. 642.

[2] Taiwo, K. "Dimensions and issues of mobile agent technology." *Villanova Journal of Science, Technology and Management, Nigeria*. 2019; 72–86.

[3] Al-Shayea, T. K., Mavromoustakis, C. X., Batalla, J. M., and Mastorakis, G. "A hybridized methodology of different wavelet transformations targeting medical images in IoT infrastructure." *Measurement*. 2019. p. 148.

[4] Al-Shayea, T. K., Batalla, J. M., Mavromoustakis, C. X., and Mastorakis, G. "Embedded dynamic modification for efficient watermarking using different medical inputs in IoT." *24th International Workshop on Computer Aided Modeling and Design*. IEEE; 2019.

[5] Al-Shayea, T. K., Mavromoustakis, C. X., Batalla, J. M., Mastorakis, G., and Pallis, E. "Medical image watermarking in four levels decomposition of DWT using multiple wavelets in IoT emergence." *Convergence of artificial intelligence and the Internet of things*. India: Springer; 2020. pp. 15–31.

[6] Qasim, A., Ameen, H. M. B., Aziz, Z., and Khalid, A. "Efficient performative actions for e-commerce agents." *Applied Computer Systems*, 2020; 25: 19–32.

[7] Han, Y., Zhiqi, S., Cyril, L., Chunyan, M., and Victor, L. *A survey of multi-agent trust management systems*. *IEEE Access*. 2013; 1: 35–50.

[8] Godfrey W., Jha S., and Nair S. "On a mobile agent framework for an internet of things." *International Conference on Communication Systems and Network Technologies*. India: IEEE; 2013. pp. 345–350.

[9] Toumbas, L., Mavromoustakis, C. X., Mastorakis, G., and Dobre, C. "Recent advances in remote assisted medical operations: Ambient assisted living and enhanced living environments." *Principles, Technologies and Control*. Elsevier Inc.; 2016. pp. 123–145.

[10] Dasgupta S., Aroor A., Shen F., and Lee Y. "Smartspace: Multiagent based distributed platform for semantic service discovery." *IEEE Transactions on Systems, Man, and Cybernetics:Systems*. 2014; 44(7): 805–821.

[11] Gao Q., and Cho Y. "A multi-agent personalized ontology profile based query refinement approach for information retrieval." *13th International Conference on Control, Automation and Systems*. South Korea: IEEE; 2013. pp. 543–547.

[12] Raja, A., and Raj, E. "MAD-ARM: Mobile agent based distributed association rule mining." *International Conference on Computer Communication and Informatics*. India: IEEE; 2013. pp. 1–5.

[13] Higashino, M., Hayakawa, T., Takahashi, K., Kawamura, T., and Sugahara, K. "Management of streaming multimedia content using mobile agent technology on pure P2P-based distributed e-learning system." *27th International Conference on Advanced Information Networking and Applications*. Spain: IEEE; 2013. pp. 1041–1047.